PATENT ABSTRACTS OF JAPAN

(11)Publication numb r : 06-068049
(43)Date of publication of application : 11.03.1994

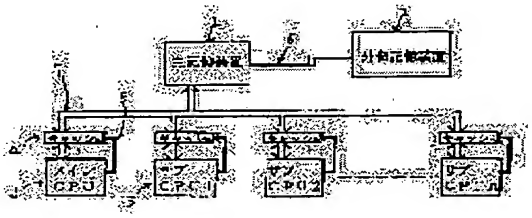
(51)Int.Cl. G06F 15/16
G06F 9/44

(21)Application number : 04-221188
(22)Date of filing : 20.08.1992

(71)Applicant : SHARP CORP
(72)Inventor : OGAWA KENJI

(54) COMPUTER ARCHITECTURE

(57)Abstract:
PURPOSE: To increase the execution speed and to easily perform the control by optimizing object assignment to each CPU.
CONSTITUTION: A main CPU 3 successively assigns sets of objects of high independence so that sub-CPUs 5 are loaded as equally as possible. Each sub-CPU 5 successively executes the assigned object group and outputs flags indicating the end of individual objects and instance variables required for objects in the other sub-CPUs 5 in accordance with requests of the main CPU 3. The main CPU 3 monitors flags indicating the end of objects in sub-CPUs 5 and adjusts the timing for the purpose of preparing for execution of mutually related objects in different. sub-CPUs 5 and delivers the instance variables. Thus, a series of operations are repeated till the end of all objects.



LEGAL STATUS

[Date of request for examination]

[Date of sending the examiner's decision of rejection]

[Kind of final disposal of application other than the examiner's decision of rejection or application converted registration]

[Date of final disposal for application]

[Patent number]

[Date of registration]

[Number of appeal against examiner's decision of rejection]

[Date of requesting appeal against examiner's decision of rejection]

[Date of extinction of right]

(19)日本国特許庁（ＪＰ）　(12)**公開特許公報（Ａ）**　(11)特許出願公開番号

特開平6－68049

(43)公開日　平成６年(1994)３月11日

(51)Int.Cl.[5]	識別記号	庁内整理番号	ＦＩ	技術表示箇所
Ｇ０６Ｆ　15/16	３７０　Ｚ	8840－5L		
9/44	３３０　Ｚ	9193－5B		

審査請求　未請求　請求項の数１(全　５　頁)

(21)出願番号　特願平4－221188

(22)出願日　平成４年(1992)８月20日

(71)出願人　000005049
シャープ株式会社
大阪府大阪市阿倍野区長池町22番22号

(72)発明者　小河　健治
大阪府大阪市阿倍野区長池町22番22号　シャープ株式会社内

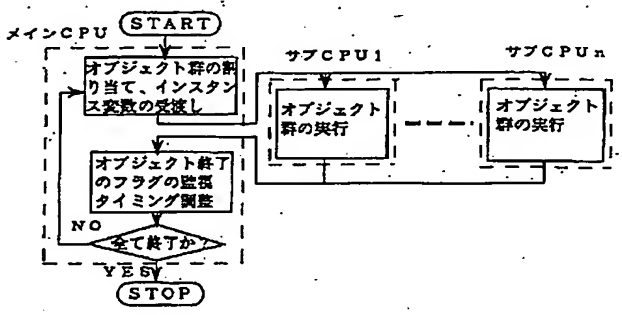
(74)代理人　弁理士　梅田　勝

(54)【発明の名称】　コンピュータ・アーキテクチャ

(57)【要約】

【目的】　本発明のコンピュータ・アーキテクチャは、オブジェクト指向プログラムを実行するコンピュータにおいて、高速化と効率化を図る。

【構成】　メインＣＰＵが、それぞれ独立性が高いオブジェクトの集まりを、各サブＣＰＵの負荷が出来るだけ均等になるように順次割り当てる。



1

【特許請求の範囲】

【請求項１】　各ＣＰＵに置かれ一時的な記憶をするキャッシュメモリと、記憶手段を有する主記憶手段と、データバス，アドレスバス，その他周辺装置からなるコンピュータ・ハードウェアと、

ＣＰＵの内オブジェクト群を順次実行する複数のサブＣＰＵと、

オブジェクト指向のプログラムを実行するために各装置への制御を行う制御手段と、前記サブＣＰＵへのオブジェクト群の割り当てを行う分割手段と、オブジェクトの終了を示すフラグの監視を行う監視手段と、タイミングの調整を行う調整手段と、オブジェクト間のインスタンス変数の受け渡しを行う受け渡し手段とを有するメインＣＰＵと、

プログラム実行前に前記メインＣＰＵの処理の組み立ても含んだコード変換を行うコンパイラとからなることを特徴とするコンピュータ・アーキテクチャ。

【発明の詳細な説明】

【０００１】

【産業上の利用分野】本発明はコンピュータ・アーキテクチャに関するものである。

【０００２】

【従来の技術】従来の技術によるコンピュータのアーキテクチャは図４に示すように、１個のＣＰＵと主記憶装置と命令・データバス、アドレスバス、その他周辺装置より構成されるもので、プログラムを記憶装置に記憶させて１個ずつの命令を逐次実行する。また、ＣＰＵと主記憶装置の間に高速なキャッシュ・メモリ（緩衝記憶装置）を設けている。

【０００３】図４のシングルプロセッサ型のコンピュータはプログラムの実行速度が本質的にＣＰＵの実行速度により制約されている。

【０００４】高速化が要求される場合には、図５のように、複数のＣＰＵと複数のキャッシュメモリと主記憶装置および命令・データバス、アドレスバスその他周辺装置より構成されるマルチプロセッサ型のコンピュータが採用される。マルチプロセッサ型のコンピュータは、プログラムを適当な大きさのモジュールに分割して複数のＣＰＵで実行させる事により高速化を図っている。

【０００５】しかし、マルチプロセッサ型のコンピュータは、ＣＰＵの数を増やしてコストをかけた割にはそれほど性能が上がらない場合が多い。その原因は、主にモジュール分割の仕方が悪くて各ＣＰＵの動作が独立性が薄いために、各ＣＰＵから共通資源であるメモリやバスへのアクセスが頻繁に発生したり、キャッシュの不整合が発生したりするためで、極端にスピードが落ちることもある。

【０００６】キャッシュの不整合とは本来同一であるべきデータが別々のキャッシュに存在して、一方のデータがＣＰＵに書き換えられた時にデータが一致せずに矛盾

2

が起こってしまうものである。これらの事を避けて実行性能を上げるためには、なんらかの複雑な制御が必要になる。

【０００７】図６は逐次制御のコンピュータにおけるタイミング図を示す。

【０００８】プログラムとしては、これまでの主流で従来技術の逐次制御すなわちノイマン型コンピュータに適している、手続き型プログラムと、近年ユーザーインターフェースを良くしたり、プログラム開発を容易にするために開発され、今後主流になると目されている、オブジェクト指向プログラムが存在する。

【０００９】手続き型プログラムはデータと手続きが別々でサブルーチンを多用しているために、マルチプロセッサ型のコンピュータにより手続き型プログラムを実行させると独立性が高いモジュールに分割することが難しく、十分な性能が得られにくい。

【００１０】一方、オブジェクト指向プログラムではデータと手続きが一まとまりになったオブジェクトの集まりで表現され、各オブジェクト同士の結合はインスタンス変数と呼ばれる変数でなされ、各オブジェクトの関係は親と子といった階層構造になっている。しかし、オブジェクト指向プログラムをマルチプロセッサ型のコンピュータにより実行させる場合においても、各ＣＰＵへの最適なオブジェクトの割り当てがなされていないと、各ＣＰＵから共通資源であるメモリやバスへのアクセスが頻繁に発生して十分な性能が得られるとは限らない。あるいは、なんらかの複雑な制御が必要になったりする。

【００１１】

【発明が解決しようとする課題】上述のように従来は、オブジェクト指向のプログラムをコンピュータで実行する場合に、より高速化が要求されるためにマルチプロセッサ型のコンピュータを利用しても、各ＣＰＵへの最適なオブジェクトの割り当てがなされていないと、各ＣＰＵから共通資源であるメモリやバスへのアクセスが頻繁に発生して十分な性能が得られるとは限らない。

【００１２】本発明はこのような点に鑑みなされたものであり、オブジェクト指向プログラムを実行するコンピュータにおいて、高速化と効率化を図る事を目的とする。

【００１３】

【課題を解決するための手段】本発明によれば、各ＣＰＵに置かれ一時的な記憶をするキャッシュメモリと、記憶手段を有する主記憶手段と、データバス，アドレスバス，その他周辺装置からなるコンピュータ・ハードウェアと、ＣＰＵの内オブジェクト群を順次実行する複数のサブＣＰＵと、オブジェクト指向のプログラムを実行するために各装置への制御を行う制御手段と、前記サブＣＰＵへのオブジェクト群の割り当てを行う分割手段と、オブジェクトの終了を示すフラグの監視を行う監視手段と、タイミングの調整を行う調整手段と、オブジェクト

間のインスタンス変数の受け渡しを行う受け渡し手段とを有するメインＣＰＵと、プログラム実行前に前記メインＣＰＵの処理の組み立ても含んだコード変換を行うコンパイラとからなることを特徴とするコンピュータ・アーキテクチャである。

【００１４】

【作用】上記のコンピュータ・アーキテクチャによって、従来技術によってオブジェクト指向プログラムをマルチプロセッサ型のコンピュータで実行する場合と比べて、各ＣＰＵへのオブジェクトの割り当てが最適であるので、実行速度が優れたり、制御が容易になったりする。

【００１５】

【実施例】図１に本発明のコンピュータ・アーキテクチャのブロック図を、図２にタイミング図を示している。また、図３はオブジェクト指向プログラムの流れ図の簡単な例を示す。

【００１６】オブジェクト指向プログラムの場合、図３の流れ図のように例えば図形を描くオブジェクト１や、ウインドウタイトルを出力するオブジェクト２、テキストを表示するオブジェクト３等が同時に、独立して実行される場合が多い。

【００１７】本発明のコンピュータ・アーキテクチャは、メインＣＰＵが、それぞれ独立性が高いオブジェクトの集まりを各サブＣＰＵの負荷が出来るだけ均等になるように順次割り当てる。サブＣＰＵは割り当てられたオブジェクト群を順次実行し、メインＣＰＵの要求に応じて個々のオブジェクトの終了を示すフラグや他のサブＣＰＵ内のオブジェクトが必要とするインスタンス変数を出力する。また、メインＣＰＵはサブＣＰＵのオブジェクトの終了を示すフラグを監視したり、関連のあるオブジェクトどうしが別々のサブＣＰＵで実行される場合などのためにタイミングを調整したり、インスタンス変数の受け渡しをしたりする。これら一連の動作を全てのオブジェクトが終了するまで繰り返す。

【００１８】メインＣＰＵの動作は、あらかじめプログラムの内容に応じてコンパイラによって最適に組み立てられる。

【００１９】なお、メモリ領域は従来技術でも行われている仮想メモリの仕組みで主記憶装置だけでなく、周辺

装置の外部記憶装置に取られる事もある。

【００２０】またサブＣＰＵの数は、要求性能とコストにより決められる。

【００２１】従来技術においてもＣＰＵを複数個設けたマルチプロセッサ型のコンピュータが存在するが、本発明の方がよりオブジェクト指向プログラムの特徴を生かしたアーキテクチャを採用したために実行速度がすぐれていたり、制御が容易であったりする。

【００２２】

【発明の効果】以上のように本発明によれば、従来技術のコンピュータよりも実行速度が優れていたり、制御が容易であったりする。特に、各サブＣＰＵが独立性の高いオブジェクトを実行するので、メモリをアクセスしなくてもキャッシュメモリやＣＰＵ内の高速なレジスタに必要なデータが見つかる事が多いので実行速度を速くすることができる。

【図面の簡単な説明】

【図１】本発明のコンピュータ・アーキテクチャが適用されるコンピュータのブロック図である。

【図２】本発明のコンピュータ・アーキテクチャが適用されるコンピュータのタイミング図である。

【図３】本発明のコンピュータ・アーキテクチャのオブジェクト指向プログラムの流れを示す図である。

【図４】従来技術におけるシングルプロセッサ型の構成を示すブロック図である。

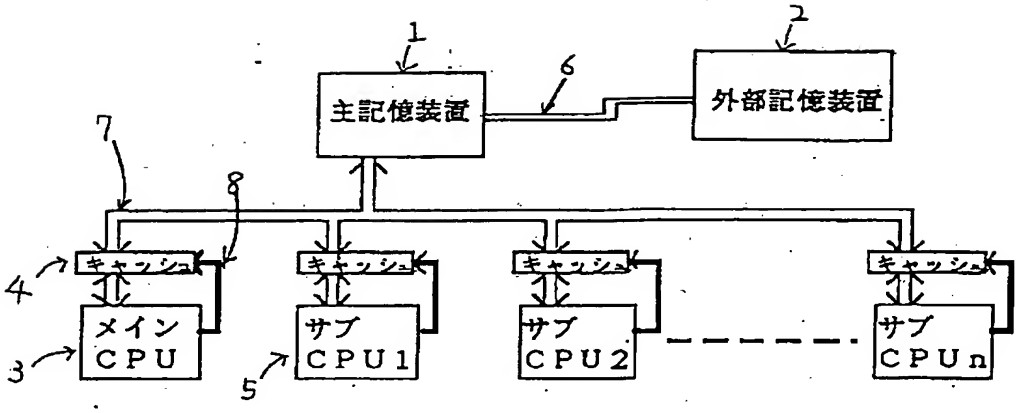
【図５】従来技術におけるマルチプロセッサの構成を示すブロック図である。

【図６】従来技術での逐次制御のコンピュータにおけるタイミング図である。

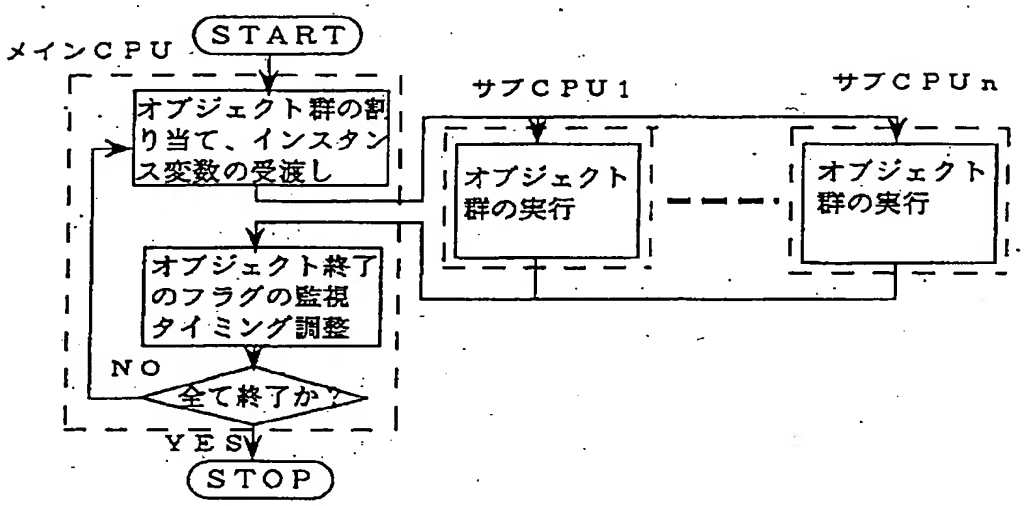
【符号の説明】

１　主記憶装置
２　外部記憶装置
３　メインＣＰＵ
４　キャッシュ・メモリ
５　サブＣＰＵ
６　メモリバス
７　命令・データバス
８　アドレスバス
９　ＣＰＵ

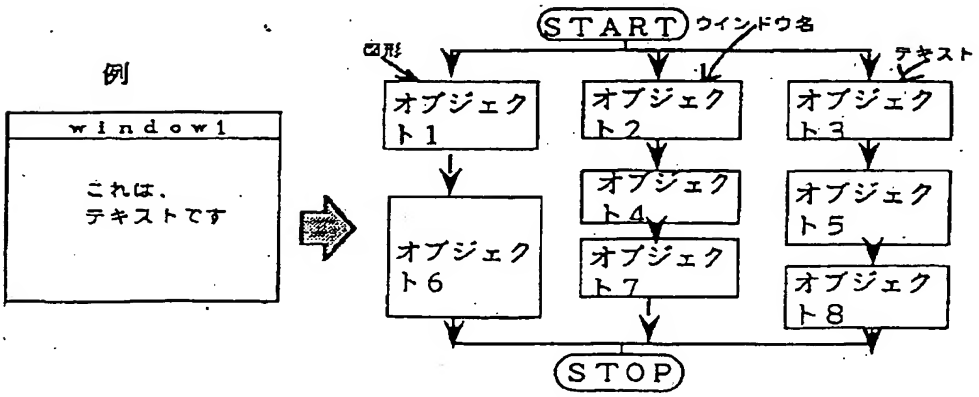
【図１】


1
2
6
主記憶装置
外部記憶装置
7
8
4
3
5
キャッシュ
キャッシュ
キャッシュ
キャッシュ
メイン
CPU
サブ
CPU1
サブ
CPU2
サブ
CPUn

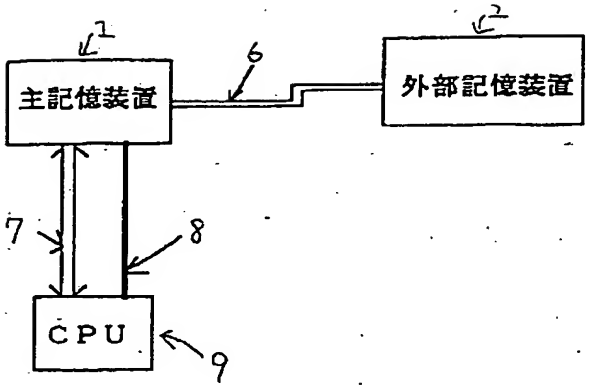
【図２】


メインCPU
START
オブジェクト群の割り当て、インスタンス変数の受渡し
オブジェクト終了のフラグの監視タイミング調整
NO
全て終了か？
YES
STOP
サブCPU1
オブジェクト群の実行
サブCPUn
オブジェクト群の実行

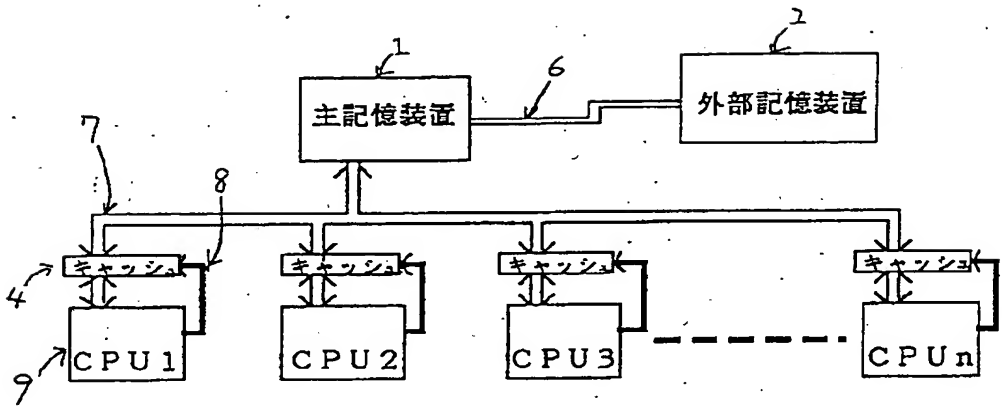
【図３】


例
window1
これは、
テキストです
START
ウインドウ名
図形
テキスト
オブジェクト1
オブジェクト2
オブジェクト3
オブジェクト4
オブジェクト5
オブジェクト6
オブジェクト7
オブジェクト8
STOP

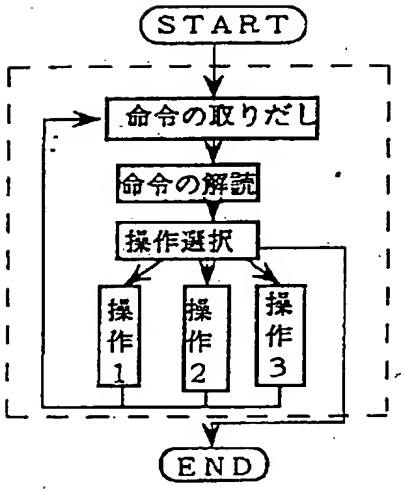
【図4】


1
2
主記憶装置
6
外部記憶装置
7
8
CPU
9

【図5】


1
2
主記憶装置
6
外部記憶装置
7
8
キャッシュ
キャッシュ
キャッシュ
キャッシュ
4
9
CPU1
CPU2
CPU3
CPUn

【図6】


START
命令の取りだし
命令の解読
操作選択
操作1
操作2
操作3
END